

### Bit wise operators

This is one of the interesting topic.

These operations are performed on the bits of the data not on the whole data either a single unit but on the bits of the data.

operations available, bitwise AND (&) bitwise OR (|) bitwise XOR (^),

bitwise NOT (~), left shift and right shift. (<<, >>)

Bit 1	Bit2	Bit & bit2
0	0	0
1	0	0
0	1	0
1	1	1

how these operations work ?

See if there are two bits, bit one and bit 2, Two-Bits are there and both the bits are zero then bit 1 and bit 2. Then bit1&bit2 will also be zero,

And if one of the bit is one then also it is zero, if the zero this is one then the zero.

And if both are one then only it is one, so it is just like a logical AND.

So when we use a bit wise AND it will be one if both the bits are one.

Then if I say OR then this will be one.

Bit 1	Bit2	Bit   bit2
0	0	0
1	0	1
0	1	1
1	1	1

If any one of the bit is one, it will one, out of these two bit this is one that is one or if both are one so the answer is one.

## XOR

Bit 1	Bit2	Bit   bit2
0	0	0
1	0	1
0	1	1
1	1	0

Then, if both the bits are different then only it will be one. Both the bits are same then it will be zero

Xclusive OR means exclusively if one is one.

The other bit must be zero or if one bit is zero then the other bit must be one.

So it is XOR. NOT means whatever the bit is there if it is one that will become zero if it is zero it will become one than this.

Int x = 11 , y = 5 ;

X = 00001011

Y= 00000101

using examples while explaining that might also explain what the shift operations.

If I have an integer variable X and the value is 11 and the value is 5, now these two int variable we have taken we know that these numbers are actually stored in the binary form.

So what is the value of x in the binary form. 11 means 1 0 1 1.

If I say int takes two bites then it will be total 16 bits. So four bits are there so remaining all bits will be zero only. But I will not write 16 bits here I will only write only 8 bits here then same way y value five that is 1 0 1 and the remaining bits are all zeroes. this over 8 bits only and If use AND operation. to store result in one more variable z.

if y say z assign X&Y and then bits are AND 1 and 1 1 1 and 0 0 0 1

is 0 1 and 0 is also zero. So this will be one. So by binary form it is one.

So z will get the value one.

So that's how bitwise AND is perform.

So if you & 11 and 5 then the result is one. I will change the value instead of 5 will take 7.

Let us the result will be, seven is 1 1 1, then what will be the result? 1 1 1 1 1 0 1 0 1 0 0.

And all these are zeroes only. then this is how much in binary form this is one one so in decimal form it

will be 3. So 11 and 7 the result is three.

what will be the result of OR operation so take x as 11 only y as 7.

Then instead of &, And I will use OR operation then that it will be 1 0 1 1 1 1 1 0 or 1 is also 1 1 0 is also 1 and all these are zeroes only then this is in the binary form it is 1 1 1 1.

So in decimal form it is 15.  $11|7$  is 15.

So it is not like addition or subtraction directly some bitwise operations you cannot get down. So if you have some practice on this one then you may be able to get it when done

now next let us see what the result of XOR Operation, this operation will take, both are same zero, These two are different. One these two are different.

One the leading all are the zeroes. So this is 12, the result is 12, in the decimal form it is twelve. one two four eight.  $8+4$  is 12.

So the answer is 12 here, NOT. will take a character type variable x with value 5 and also and also one variable y, char takes just one byte in C++. So this X in the binary form.

It will be 5 is 1 0 1 and remaining ALL bits are zeroes. So this is the binary form of 5.

Then if I say y sign not of X then let us see what happens. Not means every binary data will change its state that is zero will become one becomes zero.

So this becomes a zero and 1 and 0. Then all these are once. So this is the compliment or not of x.

```
int main() {

#include <iostream>

using namespace std;

int a=11,b=7,c;

c=a&b;

cout<<c<<endl;

int d=11,e=7,f;
```

```
f=d|e;
```

```
cout<<f<<endl;
```

```
int g=11,h=7,i;
```

```
i=g^h;
```

```
cout<<i<<endl;
```

```
char j=5,k;
```

```
k=j<<1;
```

```
cout<<(int)k<<endl;
```

```
char l=20,m;
```

```
m=l>>1;
```

```
cout<<(int)m<<endl;
```

```
char x=5,y;
```

```
y= ~ x;
```

```
cout<<(int)y<<endl;
```

```
return 0; }
```